Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Probabilistic Discriminative Models Address the Tactile Perceptual Aliasing Problem

John Lloyd, Yijiong Lin and Nathan F. Lepora
Department of Engineering Mathematics and Bristol Robotics Laboratory, University of Bristol, U.K.
Email: {jl15313, yijiong.lin, n.lepora}@bristol.ac.uk

***Abstract*—In this paper, our aim is to highlight *Tactile Perceptual Aliasing* as a problem when using deep neural networks and other discriminative models. Perceptual aliasing will arise wherever a physical variable extracted from tactile data is subject to ambiguity between stimuli that are physically distinct. Here we address this problem using a probabilistic discriminative model implemented as a 5-component mixture density network comprised of a deep neural network that predicts the parameters of a Gaussian mixture model. We show that discriminative regression models such as deep neural networks and Gaussian process regression perform poorly on aliased data, only making accurate predictions when the sources of aliasing are removed. In contrast, the mixture density network identifies aliased data with improved prediction accuracy. The uncertain predictions of the model form patterns that are consistent with the various sources of perceptual ambiguity. In our view, perceptual aliasing will become an unavoidable issue for robot touch as the field progresses to training robots that act in uncertain and unstructured environments, such as with deep reinforcement learning.**

I. Introduction

For half a century, tactile sensing has held promise to bring human-like dexterity to robot hands and manipulators. However, there has been some degree of over-optimism about how quickly this change will take place. For example, a seminal review of Automated Tactile Sensing from 1982 concluded [1]: "Intelligent, dextrous, competent, manipulating robots will surely make their appearance and pay their way in society within the coming decade." There are various reasons why this scenario still seems out of reach nearly four decades later [2], [3]; however, the deep learning revolution [4] that has swept through computer vision [5] and is now sweeping through robotics [6] gives a renewed feeling of optimism that intelligent, dexterous robots will materialise this decade. That said, tactile sensing differs fundamentally from vision [7]. While there may be some cross-over of progress between the two modalities, tactile sensing has its own unique challenges.

One such challenge is the *Tactile Perceptual Aliasing Problem* (Fig. 1). Aliasing is a well-known issue in vision when there is ambiguity in perceiving the stimulus [8], and this is known to cause difficulties for AI algorithms such as reinforcement learning [9], [10]. To the best of our knowledge, perceptual aliasing has received no attention in the artificial tactile sensing literature. The closest body of knowledge is on the subject of tactile illusions [11], [12], where it was observed [13]: "One source of tactile illusion is clearly derived from contact mechanics... extracting the attributes of a touched object from partial knowledge of one's own tissue deformation, is a noisy and ambiguous process... and is at the root of all [tactile illusions] described thus far."

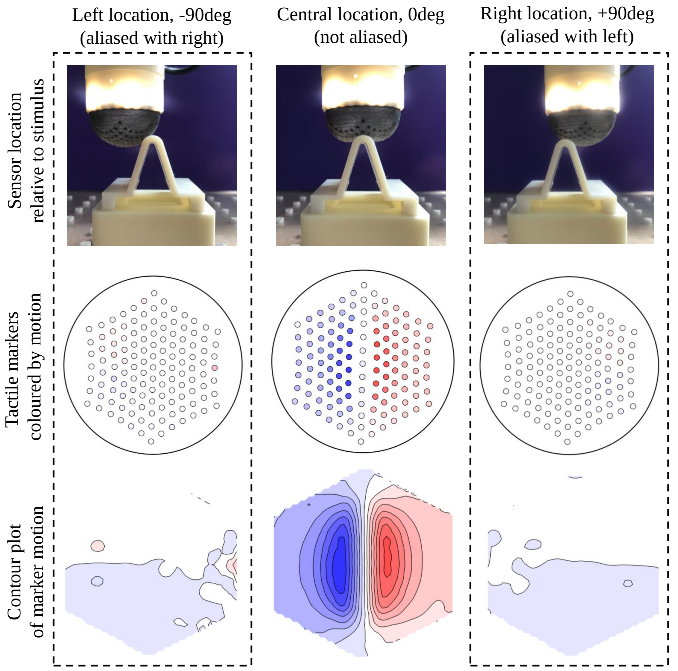

Figure 1. Tactile perceptual aliasing. The tactile sensor contacts a ridged stimulus from 3 locations (the sensor is rotated by -90 deg, 0 deg and +90 deg relative to the central location in moving between the three contacts). Markers are coloured by x-displacement for visualisation. The peripheral contacts (left, right) produce a similar sensor output even though their locations are very different, and they are both distinct from the central contact. Note that even without rotating the sensor, discriminative models can find it difficult to distinguish the weak left and right contacts from each other.

In this paper, our aim is to highlight *Tactile Perceptual Aliasing* as a problem when using deep neural networks and other discriminative models such as Gaussian process regression. The underlying cause of this problem is that tactile sensory data can be limited in its ability to disambiguate stimuli that may be physically distinct (see Fig. 1). Camera-based optical tactile sensors such as the one used in this study, offer higher-resolution tactile images than conventional taxel arrays [14], but as we will see they are still subject to significant perceptual aliasing. In our view, aliasing is a generic problem for artificial tactile sensing that will become more

This research was supported in part by a Leverhulme Trust Research Leadership award (grant number RL-2016-39 to NL).

prevalent as applications move from idealised research studies to practical robotics. Perceptual aliasing will arise whenever a physical variable extracted from a tactile image is subject to ambiguity; for example, as a contact gets lighter, the tactile imprint becomes less defined and therefore more ambiguous to localize on the sensor surface.

In future, we expect that multiple ways of dealing with tactile perceptual aliasing will be developed and used. One method that already appears to be widely used involves restricting the range over which the tactile data is collected so that ambiguous examples are excluded. Such data exclusion techniques are common in computer vision; for example, images were only selected for ImageNet if multiple human labellers agreed with each other [15]. For tactile perception of physical quantities such as edge location and orientation, recent work with deep neural networks has used training ranges that implicitly have enough contact strength to be unambiguous (*e.g.* [16, Table 1] excludes depths <1 mm). Clearly, there are many issues with the data selection method: one is the need for human intervention to fine-tune the data collection; another is the difficulty in defining suitable ranges based on static conditions of soft, compliant sensors or objects.

Here we present an alternative way of addressing tactile perceptual aliasing using a probabilistic discriminative model that captures the distribution of the training data. The model produces both a prediction and an associated uncertainty based on the modelled distribution. It also explicitly models the distribution, and hence is more informative than methods that only produce single-point estimates of uncertainty. Regions of ambiguity in the experimental parameter space indicate underlying sources of perceptual aliasing. The specific probabilistic discriminative model that we use here relies on a deep neural network to predict the parameters of a Gaussian mixture model, resulting in an architecture known as a Mixture Density Network (MDN) [17].

The main contribution of this work is to highlight the problem of perceptual aliasing in tactile sensing, and to show how probabilistic discriminative models can address this issue by estimating prediction uncertainty and capturing the underlying probability distributions. More specifically:

1) We consider tactile perception of position, orientation and curvature under experimental conditions that introduce various sources of perceptual aliasing (*e.g.* light-contact data). Two data sets of 9,000 tapping contacts that span the range of tactile properties are used in this study.

2) We apply two popular discriminitive regression methods to this data: Gaussian process regression and deep neural networks. All model hyperparameters are tuned using Bayesian optimization to ensure a fair comparison.

3) We also apply a probabilistic discriminative method to this data: a 5-component mixture density network. Once again, the hyperparameters are tuned using Bayesian optimization.

4) We show that the discriminative regression models perform poorly on aliased data, only predicting accurately when the sources of aliasing are removed. In contrast, the MDN generates more accurate predictions and can identify aliased data.

II. Background

Models in tactile sensing are typically thought of in terms of the class of algorithm used or the type of information they produce; for example, a random forest classifier or support vector machine for classifying slip [18], [19] or a convolutional neural network for predicting shape-independent hardness [20] or grasp success [21]. However, from a statistical perspective, we believe there are more informative ways of viewing these models: whether they are generative or discriminative [22], [23], and the degree to which they explicitly or implicitly model the underlying probability distribution.

Generative models capture the full joint probability density function (PDF) $p(\boldsymbol{x}, \boldsymbol{y})$ over sensor output $\boldsymbol{x}$ and tactile stimulus $\boldsymbol{y}$. The distribution is usually represented as a product of the marginal and conditional PDFs $p(\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{x})$, or $p(\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y})$. This type of model is referred to as "generative" because knowledge of the joint distribution is sufficient to generate samples from the model. Once the joint distribution is known, inference can be performed using probabilistic operations such as marginalization, conditioning or Bayes' rule. However, this often requires large amounts of data, or strong assumptions such as the conditional independence assumption made in the naïve Bayes classifier. Examples of generative models used in tactile sensing have been based mainly on naive Bayes and related histogram likelihood models [24]–[27].

Discriminative models, on the other hand, only capture the conditional PDF or some underlying statistical properties of the distribution PDF [22]. This typically makes them more data-efficient, faster to train, and often more accurate, as long as it is possible to obtain enough labelled examples [23]. Consequently, this type of approach has been used far more widely than generative approaches in tactile information processing. Examples include most of the classification and regression methods that are currently in use, *e.g.* [18]–[21], and many more covered in reviews such as [28].

One can also distinguish between discriminative approaches that explicitly model the conditional PDF (probabilistic discriminative models) and discriminative approaches that model useful properties of an implicit underlying PDF (discriminative regression or classification models).

In this study, we show that for tactile perception there are benefits to be gained from explicitly modelling a more flexible form of conditional output distribution as a PDF and predicting information using that distribution. This contrasts with many other contemporary approaches that assume (often implicitly) a simplified output distribution such as a Gaussian.

Here we model conditional PDFs using mixture density networks. Other approaches for modelling conditional PDFs were also considered, including kernel-based models [29], conditional variational autoencoders (VAE) [30], conditional GANs [31], normalizing flows [32] and neural autoregressive methods that factorize the output distribution [33]. However, like many other non-parametric approaches, kernel-based methods tend not to scale well to the large data sets that are now common in tactile sensing. Moreover, while other more complex 'conditional-generative' methods are good at modelling the underlying data generation process, particularly for high-dimensional spaces, they tend to require computationally-expensive numerical integration or optimization to extract the associated PDF or related statistics.

III. METHODS

A. Preliminary

We model the output of the tactile sensor as a continuous random vector $\boldsymbol{x} = (x_1, x_2, ..., x_M)$ that represents the output of individual tactile elements, or 'taxels'. We also model a tactile stimulus as a continuous random vector $\boldsymbol{y} = (y_1, y_2, ..., y_N)$ that represents attributes such as position, orientation and curvature. The probabilistic relationship between stimulus and sensor output is governed by a joint probability density function (PDF) $p(\boldsymbol{x}, \boldsymbol{y})$. Therefore, the inference process of tactile perception, where the applied stimulus that gave rise to an observed sensor output is inferred, is modelled by the conditional PDF $p(\boldsymbol{y} \mid \boldsymbol{x})$. Details of this formalism are given in the Appendix, with summaries of the model choice and their configurations presented below.

B. Model Configurations

In this study, we compare the performance of two popular discriminative regression models (Gaussian processes and neural networks) to a probabilistic discriminative model (mixture density networks). These regression models were selected because of their widespread popularity and because they have some interesting contrasting properties.

1) Gaussian Processes: The kernel type and length scale were optimized in an outer loop using Bayesian optimization [34]; parameters including the signal and noise levels were optimized in an inner loop using the built-in quasi-Newton optimizer. We tried several different types of kernel, including the exponential, Matern 3/2, Matern 5/2, rational quadratic and squared exponential kernels [35]. All other parameters and settings were left at their default values. To cope with the relatively large datasets used in this study, we experimented with several different sparse approximation methods, including Subset of Data (SoD), Subset of Regressors (SoR) and Fully-Independent Conditional (FIC) methods [36]. Of these, the SoR method consistently produced the best performance, and so we used it for all of our GP models.

2) Deep Neural Networks: The output layer is a dense (fully-connected) layer with linear activation function. All hidden layers comprise four sub-layers: dense, followed by batch normalization, activation function, then dropout. We experimented with several activation functions and obtained the best performance using Exponential Linear Units (ELUs) [37].

To improve generalization, we optimized the dropout and L2 regularization coefficients, together with the number of hidden layers and hidden layer size, in an outer loop using Bayesian optimization [34]. We found that dropout was most effective using a coefficient in the range 0 to 0.5. We applied an L2-norm regularizer to the hidden- and output-layer weights, and to the gamma parameter of the batch normalization [38] sub-layers. This regularization was found to be most effective with a coefficient in the range 0.1 to 10.0.

All neural network models were trained using the Adam optimizer [39] for 2,000 training epochs with a batch size of 256. For all models, we used a learning rate of 0.001 with an exponential decay rate of 0.5 over 10,000 decay steps.

3) Mixture Density Networks: In this study, we focus on predicting isolated tactile properties so a univariate Gaussian mixture model is used to model the output distributions. In this context, a deep neural network is used to predict the mixture weights and scalar-valued component means and standard deviations, based on the network inputs (Fig. 2).

The MDN output layer has three sub-layers, corresponding to the three different types of GMM parameter: a) the component means $\boldsymbol{\mu}_i$ are implemented using a dense layer, as for the neural network regression model; b) the mixture weights $\boldsymbol{\alpha}_i$ are implemented using a dense layer, followed by a *softmax* activation function that ensures the corresponding outputs are positive and sum to one. The standard deviations $\boldsymbol{\sigma}_i$ are implemented by a dense layer, followed by the *softplus* activation function, $f(x) = \log(1 + \exp(x))$, which ensures that the corresponding outputs are positive. We found that the *softplus* activation function tends to produce more stable training than more nonlinear alternatives (*e.g.* the *exp* function).

As for the neural network regression models, we applied batch normalization and dropout to all hidden layers. We also applied an L2-norm regularizer to the hidden and output layer weights, and to the gamma parameter of the batch normalization sub-layers, using a scale parameter in the range 0.1 to 10.0. We found that higher levels of regularization were even more beneficial for MDN models (in terms of convergence speed and stability) than for neural network regression models, possibly because of the highly nonlinear way the predicted distribution parameters interact in the loss function.

During our initial experiments, we found no significant variation in predictive performance when using different numbers of mixture components, so long as there were enough components to represent salient features of the output distribution, such as multimodality. For the type of problems considered here, we found that a five-component MDN (which we denote as MDN5) provided an effective compromise between distribution fit, data requirements and training time. All MDNs were trained using the same type of optimizer and the same default training parameters that were used to train the neural network regression models.

C. Hyperparameter Optimization

The discriminative models used in this study have several hyperparameters that affect their prediction and generalization abilities. Typically, these hyperparameters are found using simple methods such as trial-and-error, grid search, random search, or more sophisticated search algorithms such as Bayesian optimization [34]. In this study, we used Bayesian optimization to optimize the most important hyperparameters

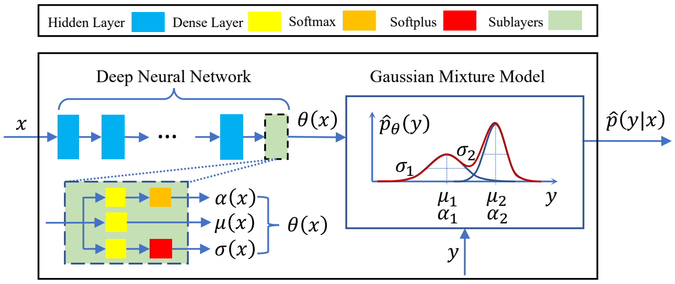

Figure 2. The mixture density network used here relies on a deep neural network to predict the parameters of a Gaussian mixure model. The neural network output layer is comprised of three parallel sub-layers (each having a different internal structure) for predicting the three different types of mixture model parameters (component weights, means and standard deviations).

for the GP, NN and MDN models (Table I) and set the remaining ones to hand-optimized or default values.

The Bayesian optimizer was configured to use the 'expected-improvement-plus' acquisition function, and run for 40 trial evaluations for each model. For each Bayesian optimization trial, the models were trained on 80% of the training data and evaluated on the remaining 20%. After the optimization process was complete, the optimized models were retrained using all of the training data.

We used the mean-squared error (MSE) between the predicted and target outputs of the validation data to evaluate the performance of the GPs and NNs, and the (conditional) negative log-likelihood of the validation data for the MDNs. During our initial experiments, we found that certain hyperparameter combinations produced particularly large and noisy error values, which led to instability and inaccuracy in the optimization process, causing it to take longer or to fail catastrophically. By way of mitigation, we found that applying the *logmod* function, $f(x) = \mathrm{sign}(x) \cdot \log(1 + |x|)$, to the error function helps compress extremely large and small values to reduce these problems. Since *logmod* is a monotonically-increasing function, the optima of the transformed loss lie at the same values as the original loss.

D. Robotic System and Software Environment

The robotic system used for this study consisted of an optical-based tactile fingertip sensor (the TacTip), mounted as an end-effector on an IRB 120 robot arm (ABB Robotics).

Table I
MODEL HYPERPARAMETERS FOR OPTIMIZATION

Model	Hyperparameter	Value Range
GP	Kernel Type	{Exponetial, Matern 3/2, Matern 5/2, Rational Quadratic, Squared Exponential}
	Kernel Scale	[1e-4, max(1e-3, 10σ_y)]
NN/MDN	Hidden Layers	{1,2,3,…,10}
	Hidden Units	{16,32, 64,…512}
	Dropout	[0,0.5]
	L2 Regularizer	[0.1, 10] (log transformed)

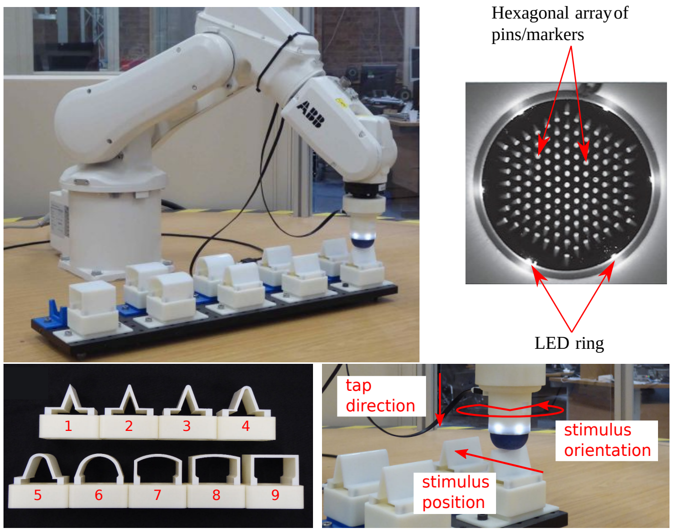

Figure 3. Robotic system used to capture the tactile data. The TacTip sensor is mounted as an end-effector on an IRB 120 robot arm and used to record tactile data from a set of 3D-printed stimuli mounted on a fixed plate. A hexagonal array of pins on the inside surface of the tip transduces and amplifies deformation of the skin surface. The nine edge-based stimuli span a range of curvatures, varying from a sharp edge to a flat surface.

1) Tactile Fingertip: We gathered tactile data using a TacTip optical tactile sensor developed at Bristol Robotics Laboratory that has since been distributed to other laboratories [40], [41]. The 3D-printed sensor used in this paper has a hemispherical, gel-filled rubber tip with 127 internal marker pins, arranged in a hexagonal array with a pin spacing of approximately 3 mm. The markers are illuminated by an LED ring situated around the circumference of the tip. When a mechanical stimulus is applied to the tip, the pins are displaced within the gel, and this movement is captured as a sequence of 680×480-pixel images sampled at approximately 30 fps.

2) Robot Arm: The TacTip was mounted as an end-effector on a six degree-of-freedom (DOF) IRB 120 robotic arm that can position the sensor with an absolute repeatability of approximately 0.1 mm (Fig. 3).

3) Integrated Tactile Sensing and Robot Control: A modular software framework was used to control the robot arm and capture data from the tactile sensor. All high-level control and data processing was implemented using MATLAB. The robot arm was controlled using a Python client that communicates with a RAPID server running on the robot controller. The tactile sensor interface also performs several low-level image-processing functions for detecting and tracking the markers in the sensor image.

E. Dataset Collection and Preprocessing

This study focuses on tactile perception of position, orientation and curvature of ridged stimuli (Fig 3, bottom left). We also aim to generalize across different combinations of these properties so that, for example, we can predict the orientation of a stimulus across a range of positions and curvatures. For these experiments, we used a set of nine 3D-printed ridged stimuli that span a range of curvatures. The curvature of each stimulus was represented using the Logarithm of the Radius of Curvature (LROC). Except for the 1st and 9th stimuli, the radius of curvature of all other stimuli increases by a factor of two across the range, and so the LROC scales results in a linear prediction scale.

The data sets were generated by uniform random sampling 1,000 points over the ±15 mm range of positions and ±90-degree range of orientations for each stimulus. For taps within this position range, all stimuli contacted the sensor in the ±10 mm range, but only Stimuli 5-9 contacted the sensor between -10 mm and -15 mm, and between 10 mm and 15 mm. This process resulted in datasets containing 9,000 samples. Two separate datasets were gathered: the first for training and hyperparameter optimization and the second for testing. For each tactile sensing sample, the sensor was positioned approximately 1 mm above the stimulus and then tapped down 5 mm. Then five consecutive frames were recorded before returning the sensor to the start position. Each set of frames was labelled with the stimulus position, orientation and curvature.

Before being used as an input to a prediction model, each sensor frame was pre-processed in OpenCV, using an adaptive threshold and blob detection routine to produce a 254-element vector of x and y marker coordinates. A nearest-neighbour tracking routine was also used to ensure that the ordering of the 254-element vector remained consistent across datasets. The initial marker positions, recorded when the sensor was not in contact, were subtracted from the marker positions returned by the sensor to produce a vector of relative marker displacements. The five samples collected during each tap were then averaged to produce a single 254-element observation vector. We found that collecting five samples and averaging them in this way helps to remove some of the noise associated with image capture and processing.

To investigate the impact of perceptual aliasing on performance, we generated two additional train/test pairs of datasets from the original pair. The first additional pair was generated by filtering out the samples associated with flat Stimulus 9 and all samples lying outside the ±10 mm position range. This pair of filtered datasets did not include any of the potential aliasing effects associated with indistinguishable free-space taps on either side of Stimuli 1-4, or indistinguishable taps across the flat Stimulus 9, and was used to exclude aliasing effects for position and orientation prediction.

The second additional pair of datasets was generated by filtering out the samples for Stimuli 1-3 and all taps lying outside the ±5 mm position range. This pair of filtered datasets did not include any of the potential aliasing effects due to the TacTip not being sufficiently compliant to discriminate between the sharper edges of Stimuli 1-3, or it being unable to discriminate between curvatures in positions where there is minimal contact with the sensor. This second additional pair of datasets was used to exclude aliasing effects for curvature prediction. In this study we refer to the original pair of datasets as the *aliasing* datasets, and the additional filtered pairs of datasets as the *no-aliasing* datasets.

IV. Results

A. Predictive accuracy of discriminative regression models

Two types of discriminative regression model were trained to predict the position, orientation and curvature of stimuli, based on the tactile sensor observations: Gaussian processes and deep neural networks. In each case, the model hyperparameters were optimized as described in the Methods (see Appendix, Table IIIa,b). The model performance is summarized by the mean and standard deviation of the RMS test errors (Table II).

The predictive accuracy of both models is significantly degraded when perceptual aliasing is present in the data (Fig. 4, *c.f.* left and right pairs of columns). This is indicated by the larger RMS errors and wider dispersion of predictions around the red ground-truth line for the aliasing dataset when compared to the no-aliasing dataset, which is confirmed in the overall RMS test errors (Table II). It is also clear that the NN models produce a lower RMS error and narrower distribution around the ground-truth line than the GP models, so they are more accurate for this set of tasks. Note also that both models are only capable of producing single-point predictions and do not estimate prediction uncertainties (although a standard GP can estimate the predictive variance of an output variable, this only represents how confident it is about its prediction of the output; it does not model the distribution of noise around the output).

B. Predictive accuracy of probabilistic discriminitive models versus discriminative regression models

A similar analysis to that described in Section IV-A was applied to a 5-component mixture density network (MDN5), as an example of a probabilistic discriminative model. As above, the model hyperparameters were optimized as described in the Methods (see Appendix, Table IIIc). For all MDN models, the predictions of position, orientation and curvature are all based on the location of the (major) mode of the predicted output distribution (see Appendix D). The model performance is again summarized by the mean and standard deviations of its RMS test errors over ten runs (Table II).

The predictive accuracy of the MDN5 model was compared to the GP and NN models on the aliasing and no-aliasing tactile datasets (Table II). For each training run on the no-aliasing dataset, the RMS error was computed using the no-aliasing test set. For each training run on the aliasing dataset, the RMS error was computed using *both* the aliasing test set and the no-aliasing test set. The error on the no-aliasing test set indicates the residual error that would occur if it were possible to identify and remove the points associated with aliasing at test time.

For the no-aliasing data set, the MDN5 model is consistently more accurate than the other models (Table II, left 3 columns). The accuracy of all three models is generally better for no-aliasing data than it is for aliasing data.

For the aliasing data set, the NN model is the most accurate for position and orientation, and the MDN5 model is the most accurate for curvature (Table II, middle 3 columns).

Table II
MODEL PERFORMANCE (10 TRAINING RUNS): MEAN AND STANDARD DEVIATION RMS TEST ERRORS

Model	***No Aliasing Train / Test Data***			***Aliasing Train / Test Data***			***Aliasing Train Data / No Aliasing Test Data***		
	Pos. (mm)	***Orient. (deg)***	***Cur./LROC (log mm)***	***Pos. (mm)***	***Orient. (deg)***	***Cur./LROC (log mm)***	***Pos. (mm)***	***Orient. (deg)***	***Cur./LROC (log mm)***
GP	0.805 ± 0.006	8.443 ± 0.091	0.126 ± 0.002	4.393 ± 0.028	21.302 ± 0.307	0.828 ± 0.008	0.774 ± 0.007	8.565 ± 0.414	0.833 ± 0.008
NN	0.555 ± 0.011	3.060 ± 0.122	0.100 ± 0.001	**3.508 ± 0.137**	**9.154 ± 0.236**	0.645 ± 0.005	0.543 ± 0.033	**1.420 ± 0.069**	0.468 ± 0.011
MDN5	**0.514 ± 0.040**	**1.310 ± 0.173**	**0.063 ± 0.048**	5.058 ± 0.481	15.731 ± 0.711	**0.592 ± 0.018**	**0.447 ± 0.053**	1.470 ± 0.184	**0.436 ± 0.034**

For the models trained using the aliasing training set and tested using the no-aliasing test set, the MDN5 models are consistently more accurate than the corresponding NN or GP models (Table II, right 3 columns), except for orientation where the NN and MDN5 models are about the same.

C. *Single-point uncertainty values using probabilistic discriminative models*

One of the benefits of using an MDN model is that it is relatively easy to compute the uncertainty associated with predictions without resorting to computationally-expensive techniques such as Monte Carlo integration or optimization. More specifically, single-point predictions can be computed (or easily approximated) from either the mean, median or mode extracted from the Gaussian mixture model, and uncertainty can be computed using the standard deviation or entropy (see Appendix D).

Two prediction/uncertainty combinations were considered for the MDN5 model (mode/entropy and mean/standard deviation) on both the aliasing and no-aliasing data sets (Fig. 5; colour indicates the level of uncertainty). In general, the uncertainty is high where predictions are further from the red ground truth line or where the spread of predictions is larger.

For the no-aliasing dataset, there is relatively little variation in prediction accuracy, with just a few high-uncertainty outliers that lie some distance from the ground-truth line (Fig. 5, left two columns). Note that the MDN model has captured the extremes of the position and orientation ranges better than the GP and NN models, with less scatter in the predictions.

For the aliasing dataset, there is a striking variation in prediction accuracy originating from several different perceptual aliasing effects (Fig. 5, right two columns):
(i) The central horizontal band for the position model (top row) corresponds to indistinguishable observations across the flat surface of Stimulus 9 where all locations ’feel’ the same. Note how the predictions have regressed to the mean of 0 mm.
(ii) The broad diagonal band of uncertain predictions around the ground truth line for orientation (middle row) appears due to the difficulty of perceiving the rotation of the flatter stimuli.
(iii) The near-central horizontal band for the orientation model (middle row) corresponds to indistinguishable observations when the sensor is close to tapping in free space and makes little or no contact with the sharper Stimuli 1-4.
(iv) The clusters at the extremes of the position range (top row) correspond to those same indistinguishable observations when the sensor makes little or no contact with Stimuli 1-4.
(v) The curvature model (bottom row) has worsening predictive accuracy and increasing uncertainty for the sharper Stimuli 1-4 as above due to little or not contact. This ambiguity is amplified by the sensor skin not being sufficiently flexible to conform to the sharpest ridged stimuli.

Additionally, we note that the entropy measure appears to distinguish between high-uncertainty perceptual aliasing contacts and low-uncertainty conventional contacts to a far greater extent than the standard deviation. In particular, the visual separation between the two classes of tactile contact is much clearer when using the mode and entropy measure than when using the mean and standard deviation.

D. *Application to sequential decision making*

To illustrate one benefit of uncertainty modelling for tactile perception, we consider the use of the MDN probabilistic discriminative model for a sequential decision making (SDM) scenario, where a sufficiently accurate decision must be made with as few sequential observations as possible.

A typical (normative) framework for optimal decision-making uses a cost function that depends on both the decision error and decision time (number of observations). An optimal solution to this type of problem takes the form of a *stopping rule* that accepts a prediction if the uncertainty is lower than a specified threshold but otherwise gathers more samples until the threshold is reached. The threshold is usually chosen or optimized to balance the cost of making errors against the cost of further sampling: a lower threshold tends to decrease the final decision error but leads to an increase in the number of rejected samples, and *vice versa* [42].

Since an MDN model can be used to make predictions and estimate the associated uncertainty, it can be used to investigate how the overall error and rejection rate varies for different uncertainty thresholds. This is illustrated for position and orientation SDM prediction tasks in Fig. 6, assuming the curvature does not change if a sample is rejected.

For both perceived quantities, the overall trend is that increasing the entropy-based uncertainty threshold results in an increase in RMS decision error, accompanied by a decrease in sample rejection rate, as expected (Fig. 6).

An interesting feature is that both the position and orientation models have a range of thresholds where there is both

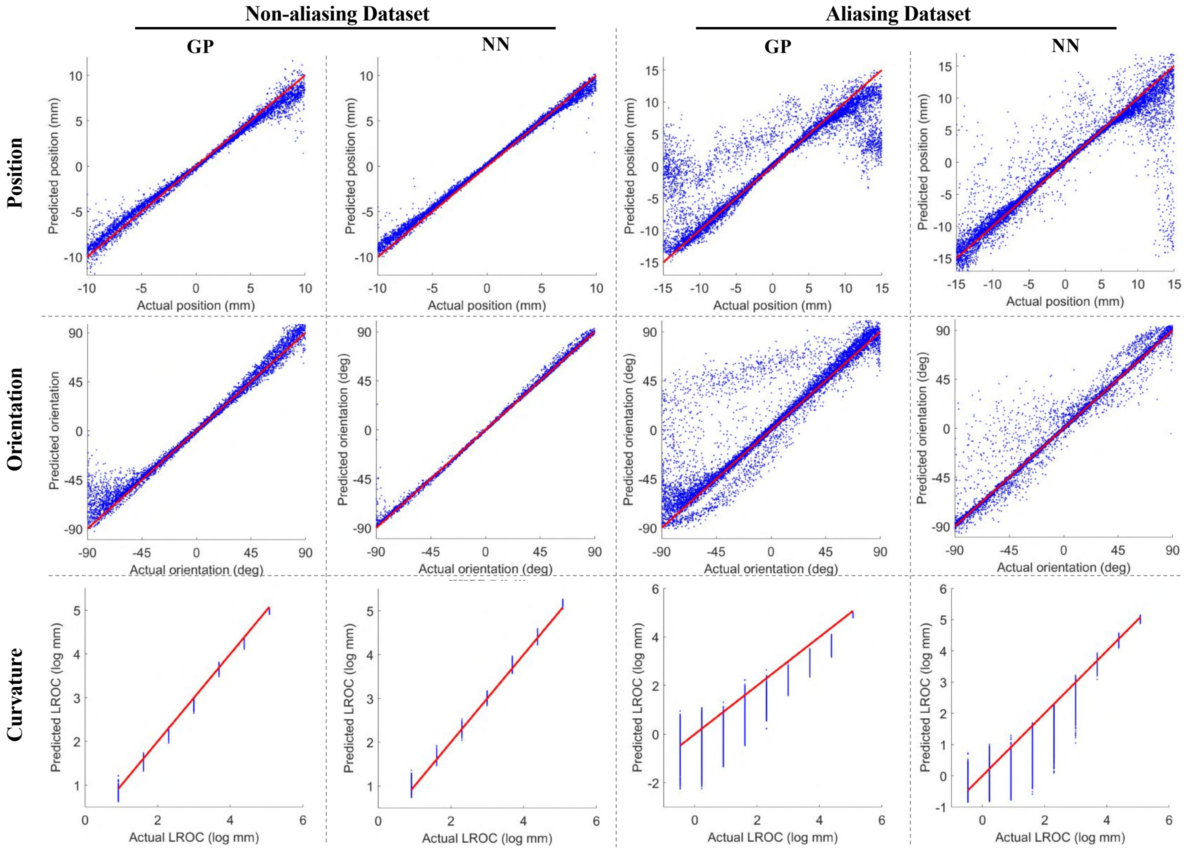

Figure 4. Results for the Gaussian process and deep neural network models trained on the no-aliasing and aliasing datasets. Predictions for position, orientation and curvature are plotted against the ground truth labels. Notice the significant scatter in the aliasing dataset with associated RMS errors in Table 2.

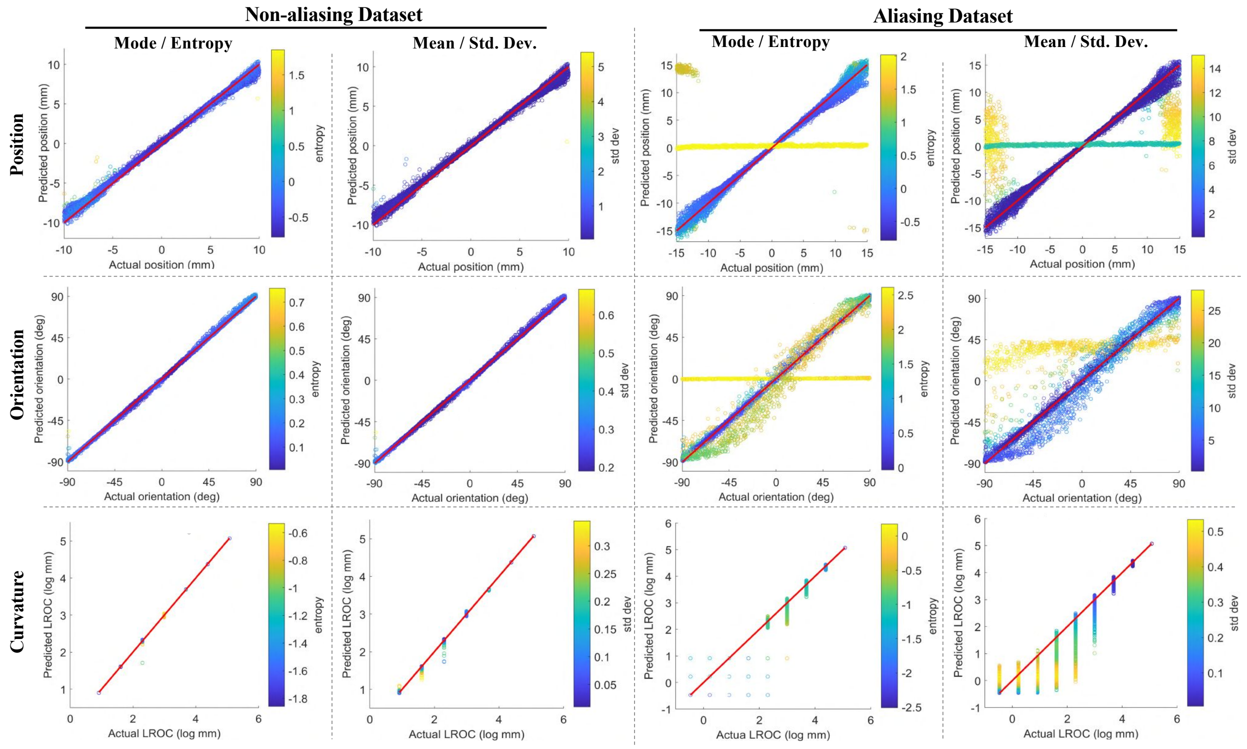

Figure 5. Results for the mixture density network model trained on the no-aliasing and aliasing datasets as in Figure 4. The prediction uncertainty is indicated by the colours of the points. Notice, again, how the high degree of scatter present in the aliasing dataset with associated RMS errors in Table 2.

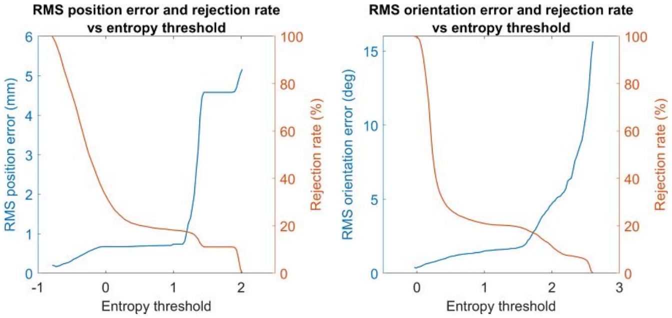

Figure 6. Sequential decision making where samples are rejected if they are above an uncertainty threshold. The MDN5 test error and rejection rate are plotted against an entropy threshold for the position and orientation models. The error improves as more samples are rejected with lower uncertainties.

relatively low RMS error and a low rejection rate (*i.e.* a 'sweet spot'). These threshold ranges lie between approximately 0.0 to 1.0 (left panel) and 0.5 to 1.5 (right panel), where the RMS errors are approximately 0.7 mm for position and 1 deg for orientation, upon rejecting about 1 in 10 samples. These accuracies are comparable with those from the non-aliasing test set (Table II), by using the model to reject ambiguous samples rather than tuning the parameter ranges.

These results could be improved further with a Bayesian approach that fuses predictions and their uncertainties over sequential observations, and with more intelligent policies for controlling the sensor to actively sample data.

V. DISCUSSION

This work shows that perceptual aliasing can be a serious issue for discriminative prediction methods such as Gaussian process regression and deep neural networks that are in common use in artificial tactile sensing. One solution is to remove the sources of aliasing, which results in improved predictive performance; however, this requires human intervention to fine-tune the data collection. We have shown instead that a probabilistic discriminative model (a mixture density network) can address this tactile perceptual aliasing problem, by modelling the distribution of the data so that the aliasing is captured. The probabilistic model gives both a prediction and its uncertainty, which can be used to identify aliased data.

We view the tactile aliasing problem as primarily one of data uncertainty, where labelled data is available to construct a predictive model, but that data has a distinctly non-Gaussian or multi-modal distribution. This contrasts with the complementary problem of model uncertainty, which arises when some of the data is missing or unavailable over part of the domain.

The ability of the mixture density network to model perceptual aliasing enables it to properly capture the distribution of tactile data. This is evident from the test results that reveal structures in the distribution, which are coloured according to the level of uncertainty (Fig. 5, right columns). In addition to being located away from the ground truth line, the uncertain predictions form patterns that are consistent with the various sources of perceptual ambiguity; for example, clusters at extreme positions for light contacts, and horizontal bands of unpredictable positions for the flat stimulus. These structures are absent from the results for the two discriminative models (Fig, 4, right two columns) because they cannot model the multi-modal, distributed nature of the perceptual aliasing.

While it is recognized that standard deep neural networks are not uncertainty-aware, one might consider that Gaussian process regression provides a useful measure of uncertainty. However, standard Gaussian processes only estimate the predictive variance, which is a measure of uncertainty in the prediction of the conditional mean, not an estimate of the conditional uncertainty in the output data distribution (they use a single global hyperparameter for the output data noise, which is optimized during model fitting). To describe perceptual aliasing, the output data uncertainty would need to be conditioned on the input variables. Heteroscedastic Gaussian processes that allow for input-dependant noise have been considered [43], [44]; however, we are not aware of how those methods could be applied to heavy-tailed or multimodal distributions which were necessary here to model tactile perceptual aliasing.

Uncertainty-aware deep learning is recognized as an important topic for robot vision. Recent work by Loquercio *et al* introduced a general framework for uncertainty estimation in deep learning [45]. Their model is based on a Bayesian belief network [46] with Monte Carlo sampling of predictions implemented using dropout [47]. While their approach captures both model and data uncertainty to some extent, it relies on propagating single-point estimates of uncertainty (assumed Gaussian at the inputs) through the model to compute single-point estimates of the mean output predictions and their uncertainties, rather than providing a full conditional distribution of the model outputs. Using that approach, it would not be possible to distinguish multi-modal distributions from unimodal distributions having the same mean and variance.

How would the uncertainty estimate be used in practice? We considered rejecting samples in a sequential decision making task if the uncertainty was too high, then re-sampling at a random location (Fig. 6). A better sampling policy would choose an action that addresses the tactile aliasing, based on knowledge contained in the model of how the uncertainty depends on sampling location relative to the object. Other information such as the sensor pose could be used in this policy, although pose would not be useful on the first contact if the object or its pose are unknown in advance. Future work in this area could relate tactile perceptual aliasing to established approaches in haptics such as active perception [48].

To conclude, the use of deep neural networks and other discriminative models bring many advantages for tactile robotic applications. However, those models are not able to capture distributed input data under tactile perceptual aliasing. In our view, perceptual aliasing will become an unavoidable issue for robot touch as the field progresses to gather training data from robots acting in uncertain and unstructured environments. Deep reinforcement learning for robot manipulation is one area where perceptual aliasing could present significant difficulties, as tactile data is more commonly adopted as a principal modality for controlling physical interaction.

References

[1] L. D. Harmon, "Automated Tactile Sensing," *The International Journal of Robotics Research*, vol. 1, no. 2, pp. 3–32, Jun. 1982, 450.

[2] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, Jun. 2019.

[3] W. Yuan, "Touch Sensing for Robots: What is the Next?" Jun. 2020. [Online]. Available: https://openreview.net/forum?id=T5ALpIJgUo2

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[6] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, Apr. 2018.

[7] V. Hayward, "Is there a 'plenhaptic' function?" *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1581, pp. 3115–3122, Nov. 2011.

[8] D. R. Williams, "Aliasing in human foveal vision," *Vision Research*, vol. 25, no. 2, pp. 195–205, Jan. 1985, 300.

[9] S. D. Whitehead and D. H. Ballard, "Learning to perceive and act by trial and error," *Machine Learning*, vol. 7, no. 1, pp. 45–83, 1991.

[10] L. Chrisman, "Reinforcement Learning with Perceptual Aliasing: The Perceptual Distinctions Approach," in *AAAI*, 1992, pp. pp. 183–188.

[11] F. de Vignemont, H. H. Ehrsson, and P. Haggard, "Bodily Illusions Modulate Tactile Perception," *Current Biology*, vol. 15, no. 14, pp. 1286–1290, Jul. 2005.

[12] V. Hayward, "A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store," *Brain Research Bulletin*, vol. 75, no. 6, pp. 742–752, Apr. 2008.

[13] ——, "Tactile illusions," *Scholarpedia*, vol. 10, no. 3, p. 8245, Mar. 2015.

[14] K. Shimonomura, "Tactile Image Sensors Employing Camera: A Review," *Sensors*, vol. 19, no. 18, p. 3933, Sep. 2019, 20.

[15] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, iSSN: 1063-6919.

[16] N. F. Lepora and J. Lloyd, "Optimal Deep Learning for Robot Touch: Training Accurate Pose Models of 3D Surfaces and Edges," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 66–77, Jun. 2020.

[17] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[18] F. Veiga, H. van Hoof, J. Peters, and T. Hermans, "Stabilizing novel objects by learning to predict tactile slip," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, Sep. 2015, pp. 5065–5072.

[19] J. W. James, N. Pestell, and N. F. Lepora, "Slip detection with a biomimetic tactile sensor," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3340–3346, 2018.

[20] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson, "Shape-independent hardness estimation using deep learning and a GelSight tactile sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 951–958.

[21] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The Feeling of Success: Does Touch Sensing Help Predict Grasp Outcomes?" in *Conference on Robot Learning*. PMLR, Oct. 2017, pp. 314–323, iSSN: 2640-3498.

[22] I. Ulusoy and C. M. Bishop, "Generative versus discriminative methods for object recognition," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 258–265.

[23] T. Jebara, *Machine learning: discriminative and generative*. Springer Science & Business Media, 2012, vol. 755.

[24] N. Jamali and C. Sammut, "Majority Voting: Material Classification by Tactile Sensing Using Surface Texture," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 508–521, Jun. 2011.

[25] H. Liu, X. Song, T. Nanayakkara, L. D. Seneviratne, and K. Althoefer, "A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1410–1415.

[26] N. F. Lepora, U. Martinez-Hernandez, and T. J. Prescott, "Active touch for robust perception under position uncertainty," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3020–3025.

[27] J. A. Fishel and G. E. Loeb, "Bayesian Exploration for Intelligent Identification of Textures," *Frontiers in Neurorobotics*, vol. 6, 2012.

[28] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, Dec. 2017, 130. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957415817301575

[29] M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya, and D. Okanohara, "Least-squares conditional density estimation," *IEICE Transactions on Information and Systems*, vol. 93, no. 3, pp. 583–594, 2010.

[30] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[32] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1530–1538.

[33] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 7184–7220, 2016.

[34] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *arXiv preprint arXiv:1206.2944*, 2012.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[36] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[37] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.

[41] N. F. Lepora, "Soft Biomimetic Optical Tactile Sensing with the TacTip: A Review," *arXiv:2105.14455 [cs]*, May 2021.

[42] A. Wald, *Sequential Analysis*. New York: John Wiley and Sons., 1947.

[43] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic Gaussian process regression," in *Proceedings of the 22nd international conference on Machine learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, Aug. 2005, pp. 489–496.

[44] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, Jun. 2007, pp. 393–400.

[45] A. Loquercio, M. Segu, and D. Scaramuzza, "A General Framework for Uncertainty Estimation in Deep Learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, Apr. 2020.

[46] B. J. Frey and G. E. Hinton, "Variational Learning in Nonlinear Gaussian Belief Networks," *Neural Computation*, vol. 11, no. 1, pp. 193–213, Jan. 1999, publisher: MIT Press.

[47] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *International Conference on Machine Learning*. PMLR, Jun. 2016, pp. 1050–1059.

[48] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, Aug. 1988.

[49] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, "On entropy approximation for gaussian mixture random vectors," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 2008, pp. 181–188.

Appendix

A. Hyperparameter optimization

Table III
Optimized hyperparameter values

(a) Gaussian process models

Response Variable	**Training Set**	**Hyperparameter**	
		Kernel Function	***Kernel Scale***
Position	No Aliasing	Rational Quadratic	17.47
	Aliasing	Rational Quadratic	6.84
Orientation	No Aliasing	Matern 5/2	21.42
	Aliasing	Rational Quadratic	9.68
Curvature / LROC	No Aliasing	Squared Exponential	18.12
	Aliasing	Rational Quadratic	23.48

(b) Neural network models

Response Variable	**Training Set**	**Hyperparameters**			
		Hidden Layers	***Hidden Units***	***Dropout***	***L2 Regularizer***
Position	No Aliasing	5	512	0.39	0.10
	Aliasing	3	256	0.13	0.10
Orientation	No Aliasing	3	512	0.46	0.10
	Aliasing	4	256	0.02	0.11
Curvature / LROC	No Aliasing	3	16	0.00	0.10
	Aliasing	10	512	0.28	0.10

(c) Mixture density network model with five components

Response Variable	**Model**	**Data Set**	**Hyperparameters**			
			Hidden Layers	***Hidden Units***	***Dropout***	***L2 Regularizer***
Position	MDN5	No Aliasing	7	256	0.38	2.27
		Aliasing	4	512	0.43	2.23
Orientation	MDN5	No Aliasing	10	512	0.02	0.57
		Aliasing	4	512	0.02	1.80
Curvature / LROC	MDN5	No Aliasing	10	64	0.09	0.10
		Aliasing	6	512	0.29	1.40

B. Probabilistic model of tactile stimulation and perception

In this paper, we model the output of the tactile sensor as a continuous random vector $\boldsymbol{x} = (x_1, x_2, ..., x_M)$ that represents the output of individual tactile elements, or 'taxels'. We also model a tactile stimulus as a continuous random vector $\boldsymbol{y} = (y_1, y_2, ..., y_N)$ that represents attributes such as position, orientation and curvature. The probabilistic relationship between stimulus and sensor output is governed by a joint PDF $p(\boldsymbol{x}, \boldsymbol{y})$. Thus, the causal forward process of transducing a stimulus into a tactile sensor output is modelled by the conditional PDF $p(\boldsymbol{x}|\boldsymbol{y})$. Similarly, the inverse inference process of tactile perception, where the applied stimulus that gave rise to an observed sensor output is inferred, is modelled by the conditional PDF $p(\boldsymbol{y}|\boldsymbol{x})$.

Following this line of reasoning, in the absence of noise we assume that the tactile sensor output $\boldsymbol{x}$ can be expressed as a deterministic, nonlinear function $\boldsymbol{x} = g(\boldsymbol{y})$ of the applied stimulus. However, due to limited accuracy, both stimulus and output may be corrupted by additive noise sources:

$$\boldsymbol{x} = g(\boldsymbol{y} + \boldsymbol{\epsilon_y}) + \boldsymbol{\epsilon}_x. \tag{1}$$

For simplicity, we assume the noises $\boldsymbol{\epsilon}_x$ and $\boldsymbol{\epsilon}_y$ are sampled from normal distributions $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_x)$ and $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_y)$ with zero means and diagonal covariance matrices $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_y$.

In tactile perception, we are usually more interested in the inverse model, which helps us infer the stimulus $\boldsymbol{y}$ that was applied to generate an observed sensor output $\boldsymbol{x}$:

$$\boldsymbol{y} = g^{-1}(\boldsymbol{x} + \boldsymbol{\epsilon}'_x) + \boldsymbol{\epsilon}'_y, \tag{2}$$

where $\boldsymbol{\epsilon}'_x = -\boldsymbol{\epsilon}_x$, and $\boldsymbol{\epsilon}'_y = -\boldsymbol{\epsilon}_y$.

Two complications can arise when using this inverse model. First, if two or more stimuli produce the same sensor output, then the sensor transformation $g(\cdot)$ might not be invertible; *i.e.* there is perceptual aliasing, which can give rise to multi-modality in the conditional distribution of $\boldsymbol{y}$. Second, even if the sensor transformation is invertible and the sensor output noise $\boldsymbol{\epsilon}'_x$ is i.i.d. Gaussian, after the noise is referred through the nonlinear function $g^{-1}(\cdot)$ its distribution will change, and the components may become dependent, asymmetric, heteroscedastic or multimodal. This problem becomes more significant where higher levels of sensor output noise $\boldsymbol{\epsilon}'_x$ are present. Therefore, an important criterion for selecting a suitable tactile perception model is whether it has sufficient representational power to capture the features of the distribution implied by Equation (2).

C. Discriminative regression models

In terms of the tactile perception (inference) problem, discriminative regression models such as Gaussian processes and neural networks assume an underlying model of the form $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}_y$, with noise $\boldsymbol{\epsilon}_y \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_Y)$. Comparing this with the model in Equation (2), we see that it implicitly assumes that: (a) the sensor transformation $g(\cdot)$ is invertible, and (b) the sensor output noise $\boldsymbol{\epsilon}_x$ is zero or negligible. If either assumption does not hold, the predictive accuracy of the model will be significantly degraded.

Gaussian Processes are a prior distribution over real-valued functions, $f \sim \mathcal{GP}(m, k)$. The prior enforces constraints on function attributes such as smoothness and amplitude. The functions $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ are the prior mean and covariance (kernel) functions. The squared exponential kernel is one of the most popular types of kernel: $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_s^2 exp(-(\boldsymbol{x} - \boldsymbol{x}')^2)/2l^2)$, where the signal level σ_s and length scale l are hyperparameters.

Once some data $\mathcal{D}$ has been observed, the prior can be replaced by a posterior distribution over functions $f \mid \mathcal{D} \sim \mathcal{GP}(m_{\mathcal{D}}, k_{\mathcal{D}})$, which can be used to predict values for new test points. Here, $m_{\mathcal{D}}(\boldsymbol{x})$ and $k_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{x'})$ are the posterior mean and covariance functions corresponding to those for the prior.

One of the problems with GPs is that they do not tend to scale well to larger datasets. This has led to the development of several sparse approximation methods that can cope with larger datasets at the expense of lower predictive accuracy.

Neural Networks are parameterised models of the form $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\varphi})$. The set of weight matrices and bias vectors constitute the model parameters $\boldsymbol{\varphi}$. The parameters are updated in an iterative manner by following the negative gradient of an error function $E(\cdot)$ that measures how close the predicted outputs $\{\hat{\boldsymbol{y}}_i\}$ are to a set of target outputs $\{\boldsymbol{y}_i\}$ for the inputs in a training data set $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$. For regression problems, the error function is usually taken to be the mean-squared error between the model outputs and targets or, equivalently, the negative conditional log-likelihood of the training data $\mathcal{D}$ under an assumed Gaussian distribution.

Within this general framework there is still a high degree of flexibility in how the model can be configured: number of hidden layers, size of each layer (number of layer outputs), activation function, optimizer type and configuration, normalization methods, regularization methods, and so on.

D. Discriminative distribution models

Unlike discriminative regression models, probabilistic discriminative models do not make strong assumptions about the underlying probabilistic model $p(\boldsymbol{y}|\boldsymbol{x})$. In fact, assuming they have sufficient representational capability, they are capable of representing any conditional PDF to an arbitrary accuracy. So, in principle, they can represent models of the form specified in Equation (2), regardless of whether the sensor transformation is invertible or the sensor output noise is negligible. However, since fitting a full distribution model generally requires more data than to fit simpler statistical functions like the mean, this increased flexibility comes at a price.

Mixture Density Networks combine the distribution modelling flexibility of a mixture model, with the parameter prediction (conditioning) power of a neural network. Although various types of mixture model can be used, in this paper we restrict our attention to finite mixtures of K multivariate Gaussian components:

$$p(\boldsymbol{y}) = \sum_{i=1}^{K} \alpha_i \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{3}$$

The parameters $(\alpha_1, \alpha_2, ..., \alpha_K)$ are the mixture weights, which are positive and sum to one; $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_K)$ are the component means, and $(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, ..., \boldsymbol{\Sigma}_K)$ are the component covariances. The mixture weights, component means and covariances are combined/flattened into a parameter vector $\boldsymbol{\theta}$, and predicted by the outputs of a multilayer neural network with parameters $\boldsymbol{\varphi}$: $\boldsymbol{\theta} = f(\boldsymbol{x}; \boldsymbol{\varphi})$. This provides the conditioning mechanism for expressing the probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ of outputs $\boldsymbol{y}$ as a function of the inputs $\boldsymbol{x}$.

The model is trained by minimizing the negative conditional log-likelihood of the training data $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)$ under an assumed Gaussian mixture distribution. Since this corresponds to a maximum likelihood (ML) approach, it is common to add a regularization penalty to the log-likelihood error to help prevent over-fitting. The minimization is carried out using the same type of stochastic gradient descent-based optimizer that is used for conventional neural networks.

Once the MDN has been trained, the mixture model parameters can be used to make predictions and estimate uncertainty. In this study, we make predictions using the conditional mean or mode of the predicted distribution. We calculate the *conditional mean* as:

$$\boldsymbol{\mu} = \sum_{i=1}^{K} \alpha_i \boldsymbol{\mu}_i. \tag{4}$$

Unfortunately, it is not possible to express the conditional (major) mode using a closed-form analytical expression. So, rather than use slower numerical methods to solve for this value, we use a faster closed-form approximation based on the mean of the highest density component. Hence, we approximate the *conditional mode* as:

$$\boldsymbol{M} = \boldsymbol{\mu}_k, \quad k = \underset{i}{\operatorname{argmax}} \left\{ \frac{\alpha_i}{\sqrt{\det(\boldsymbol{\Sigma}_i)}} \right\}. \tag{5}$$

This provides a reasonably good approximation where the components do not overlap too much. However, for other tactile perception tasks, it might make more sense to use the mean of the most probable mixture component to make predictions:

$$\boldsymbol{M} = \boldsymbol{\mu}_k, \quad k = \underset{i}{\operatorname{argmax}} \left\{ \alpha_i \right\}. \tag{6}$$

We calculate the uncertainty associated with an MDN prediction using one of two methods: we either use the 'standard deviation', which we calculate as the square root of the trace of the conditional variance $S = \sqrt{\operatorname{tr}(\boldsymbol{\Sigma})}$; or the entropy. The *conditional variance* is given by:

$$\boldsymbol{\Sigma} = \sum_{i=1}^{K} \alpha_i \boldsymbol{\Sigma}_i + \sum_{i=1}^{K} \alpha_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\mathrm{T}}. \tag{7}$$

The conditional variance $\boldsymbol{\Sigma}$ arises from two sources of variability: within-component heterogeneity associated with the first term in Equation (7), and between-component heterogeneity associated with the second.

We approximate the *conditional entropy* using a component-wise, first-order Taylor-series expansion of the logarithm of the Gaussian mixture [49]:

$$H = -\sum_{i=1}^{K} \alpha_i \log \sum_{j=1}^{K} \alpha_j \mathcal{N}(\boldsymbol{\mu}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \tag{8}$$